FERMILAB-CONF-04-459-CD

METHODOLOGIES AND TECHNIQUES FOR ANALYSIS OF NETWORK FLOW DATA

A.Bobyshev, M.Grigoriev, FNAL, Batavia, IL 60510, USA

Abstract

Network flow data gathered at the border routers and core switches is used at Fermilab for statistical analysis of traffic patterns, passive network monitoring, and estimation of network performance characteristics. Flow data is also a critical tool in the investigation of computer security incidents. Development and enhancement of flow based tools is an on-going effort. This paper describes the most recent developments in flow analysis at Fermilab.

FLOW DATA COLLECTION

At Fermilab we collect flow data from multiple network devices at the border, core and edge of the network. The total amount of daily data is about 50M records and 3-4 Gbytes in compressed binary files. All flow streams from the network devices are terminated at a single collection host that provides a short term storage space for up to 10-15 days. At this point data is also replicated in real time to several processing nodes. A few times per a hour new data from this short term storage is copied to the archiver machine that provides long term access to this data (up to 2 years) and archives data on magnetic tapes. Due to historical reasons we collect data in two different formats, the Cisco Systems Netflow Collector[1], and in the format of flow-tools[2], an open-source package for flow collection and analysis. This second package has proved to be very efficient and highly configurable for processing and analysis of a high volumes of flow data. Table 1 below gives an idea about typical amount of daily flow data we collect at each network layer.

Table 1: A daily amount of flow data

Layer	***flow-tools***	***Cisco Netflow***
Border	300 – 400 MB	0.8 – 1.6 GB
Core*	0.6 – 1.8 GB	-
Experiments** (CDF, D0)	300 – 600 MB	-

* - multiple devices, ** - per a device

TRAFFIC BREAKDOWN BY CATEGORIES

Analysis of network traffic is considered from many different perspectives to fulfil the needs of Fermilab's Data Communication Group, Computer Security Team, system administrators of the HEP collaborations and regular users. Figure 1 gives the general schema of the categories to breakdown traffic that we currently implement at Fermilab. The major categories are

- Traffic of the Fermilab's experiments or user's groups, subnets, blocks of ip addresses, i.e. clusters or farms, and individual nodes
- Application's traffic defined by the buckets of protocols, source and destination ports
- Destination traffic defined (statically or dynamically) by IP addresses, autonomous system numbers, DNS (top, second, third and so forth levels).

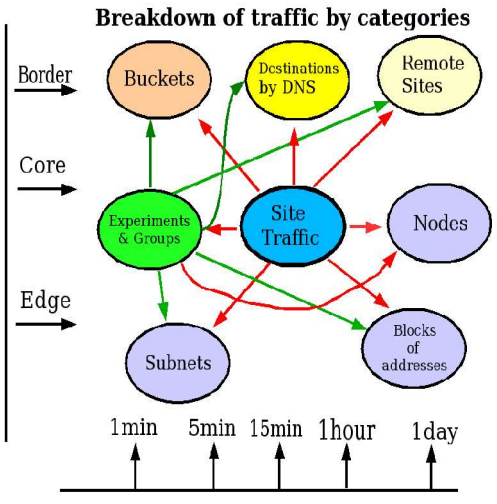


Figure 1: Scheme of traffic breakdown

The scheme above is applied at each layer, border, core and edge of network. It is also viewed in different time scopes such as:

- breakdown of traffic by periods of 1min, 5min, 15min, 1hour, 1 day and so on.
- breakdown for specified interval of time, for example, the last period of 1min, 5 min, 1 hour, and etc.
- breakdown for a time interval specified by its start and end time stamps.

THE RESULTS OF ANALYSIS.

Below we will illustrate some analysis of traffic based on flow data.

TopN Tool.

This is probably a typical application for many sites. We determine the topN senders, receivers, conversations and scanners on a hourly and daily basis and these are shown in tables available via the web. The tables show a total amount of transferred data (in MB/GB/TB), flows and packets as well as an estimate of data rate. The topN results can be seen for each logical layer of the network, the border, core or edge.

Breakdown of Traffic by experiments

This is relatively simple application that uses a flow tagging technique implemented by utilizing the flow-tag program from the flow-tools[2] package and report generator program (flow-report, from the same package). Major Fermilab experiments are identified by the static ip blocks assigned for them. An example of a daily breakdown of traffic is demonstrated in figure 2.

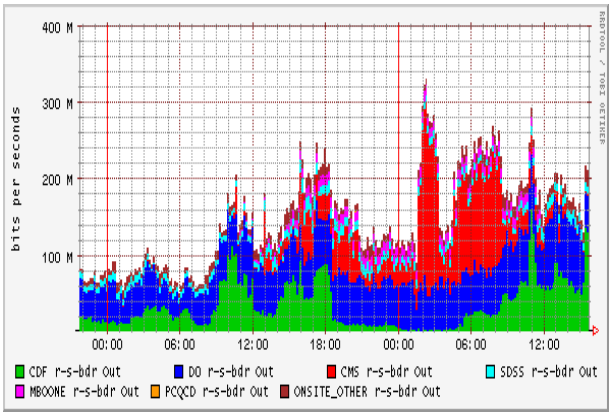


Figure 2: A daily breakdown of traffic by experiments.

Traffic Mapper.

The traffic mapper is a more complex application. It does the following tasks in near real time:

- fast name resolution of Fermilab's destinations caching into top level domain names (TLD), second, third and so forth level of DNS. This application maintains separate caches mapping destination IP addresses and networks into TLD, 2nd, 3th DNS names
- assign tags for the keys of the dynamic DNS caches, and maintain corresponding tables of symbols
- generate reports by using flow-tag and flow-report programs for previously created tagging
- use the final reports and tables of symbols to generate configurations for the dot program from graphviz package[3]
- generate graphs for predefined range of bandwidth to be shown.

The graph in figure 3 shows all connections from and to Fermilab in a 15 min interval.

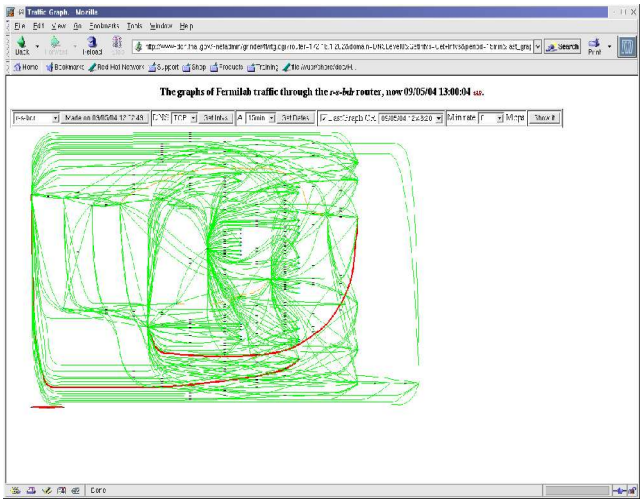


Figure 3: All network connections shown by Traffic Mapper.

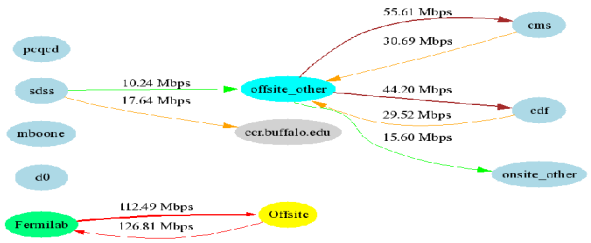


Figure 4: Traffic Map for TLDs

An interactive web-based tool allows to adjust the graph for a number of nodes to be shown, a time interval, a

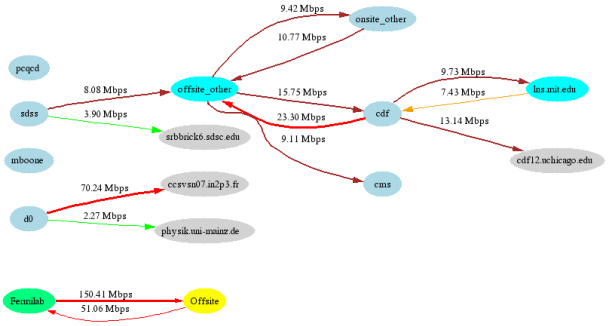


Figure 5: Example of adjusted mapper's graphs.

level of DNS, a minimal data rates to be shown.

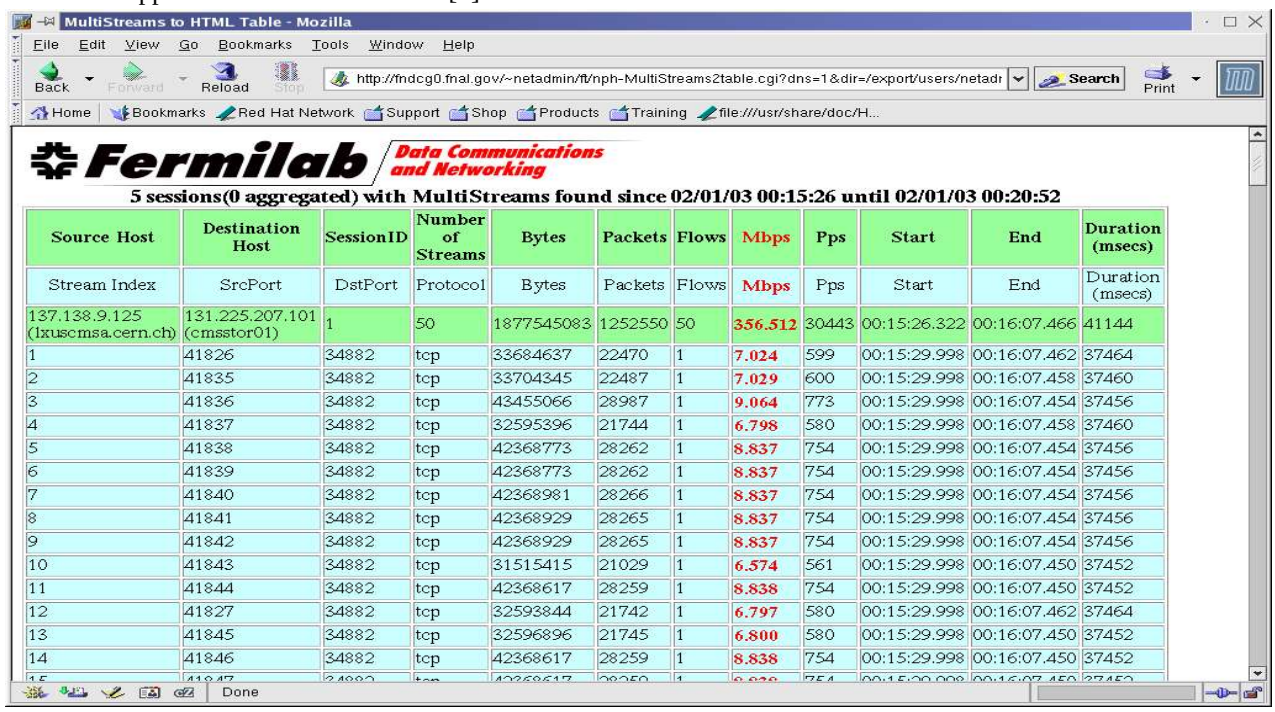
Examples of such graphs are in figure 4 and 5 above.

Detection of GridFTP

The last demonstration that we are going to introduce in this brief article is a program to estimate throughput of multistream applications such as GridFTP[4]. This tool is actively used by HEP community for GRID [4] intensive high performance data transfers from and to Fermilab. Passive monitoring of transfer performance in this case is very useful because GridFTP by design has limited capabilities for self monitoring. Monitoring based on flow information is also challenging because it has no predefined tcp ports that may be used to identify the sessions, and one assigned dynamically could be changed in time. We use the following approach to estimate GridFTP performance:

- create a hash of all flows with a key srcIP:dstIP, and records srcIP, dstIP,startclock , endclock, protocol, source and destination tcp ports for all such records.
- then we select all flows created at approximately the same time (defined by the threshold parameter in the range 1-20 secs.). For this time difference in creation of the flows we select all records with the same destination tcp port. The source ports are different for the multistream sessions.

Estimates of performance characteristics are shown for aggregated sessions and on per a stream basis. If specified we also can do aggregation of multiple flow chunks that belong to transmission of same file. For this we calculate signature of the flows, which is a digest built on sorted lists of the source ports and destination ports. Figure 6 introduces a fragment of web page with results of detection and performance estimates.



Source Host	Destination Host	SessionID	Number of Streams	Bytes	Packets	Flows	Mbps	Pps	Start	End	Duration (msecs)
Stream Index	SrcPort	DstPort	Protocol	Bytes	Packets	Flows	Mbps	Pps	Start	End	Duration (msecs)
137.138.9.125 (lxuscmsa.cern.ch)	131.225.207.101 (cmsstor01)	1	50	1877545083	1252550	50	356.512	30443	00:15:26.322	00:16:07.466	41144
1	41826	34882	tcp	33684637	22470	1	7.024	599	00:15:29.998	00:16:07.462	37464
2	41835	34882	tcp	33704345	22487	1	7.029	600	00:15:29.998	00:16:07.458	37460
3	41836	34882	tcp	43455066	28987	1	9.064	773	00:15:29.998	00:16:07.454	37456
4	41837	34882	tcp	32595396	21744	1	6.798	580	00:15:29.998	00:16:07.458	37460
5	41838	34882	tcp	42368773	28262	1	8.837	754	00:15:29.998	00:16:07.454	37456
6	41839	34882	tcp	42368773	28262	1	8.837	754	00:15:29.998	00:16:07.454	37456
7	41840	34882	tcp	42368981	28266	1	8.837	754	00:15:29.998	00:16:07.454	37456
8	41841	34882	tcp	42368929	28265	1	8.837	754	00:15:29.998	00:16:07.454	37456
9	41842	34882	tcp	42368929	28265	1	8.837	754	00:15:29.998	00:16:07.454	37456
10	41843	34882	tcp	31515415	21029	1	6.574	561	00:15:29.998	00:16:07.450	37452
11	41844	34882	tcp	42368617	28259	1	8.838	754	00:15:29.998	00:16:07.450	37452
12	41827	34882	tcp	32593844	21742	1	6.797	580	00:15:29.998	00:16:07.462	37464
13	41845	34882	tcp	32596896	21745	1	6.800	580	00:15:29.998	00:16:07.450	37452
14	41846	34882	tcp	42368617	28259	1	8.838	754	00:15:29.998	00:16:07.450	37452

Figure 6: The results of detection and performance estimates for GridFTP

ACKNOWLEDGEMENTS

Our recent efforts on flow based analysis at Fermilab is based on the flow-tool package. We would like to thank Mark Fullmer <maf@splintered.net> as well as all other contributors in development of this great tool.

Also we would like to thank all authors of the GraphViz software for a very powerful package for creation of a visualization for different data.

REFERENCES

[1] Cisco Systems Inc. Cisco CNS NETFLOW Collection Engine http://www.cisco.com

[2] The OHIO State University , Flow-tools, http://www.splintered.net/sw/flow-tools

[3] AT&T Labs-Research, Graphviz http://www.graphviz.org

[4] Globus Toolkit, http://www.globus.org

[5] Cricket,The open-source monitoring package, http://cricket.sourceforge.net/